

Index

** operator (exponentiation), 30

A

%abs, 85–86

numeric return data type and, 13*t*

absolute values. *See* %abs

accumulation operators, 83

Acos, 24

ACTGRPDFN*, 69

activation groups, 65–70, **65**

CALLER*, 68

CEETREC API and, 69

cleaning up, using RCLACTGRP, 69

default, 66

displaying, for a job, 70

ending of, 66

named, 67

NEW*, 67, 68

Original Program Model (OPM) programs
and, 66

performance considerations for, 69

scoping overrides and, ACTGRPDFN*, 69

scoping resources with OVRSCOPE and,
68–69

%addr, 86–87

definition specifications using, 9

pointer return data type and, 16*t*

address, storage. *See* %addr

addresses, procedure, retrieve. *See* %paddr

%alloc, 4, 88

pointer return data type and, 16*t*

allocating storage. *See* %alloc

AND, bitwise (%bitand), 88–89

API interfacing, 63–65

binding and, 64

CEEHDLR API and, 64–65

CEEHDLU API and, 64

dynamic calls to, 63–65

QCMDEXC, 63

Application Program Interfaces. *See* API

interfacing

arguments, 4. *See also* parameters

arrays

elements in, retrieve (%elem), 111

lookups in (%lookupxx), 124–126

substrings in (%subarr), 144–145

sum of elements in (%xfoot), 157

Asin, 24

assignment statements, indicator data type and, 11

Atan/Atan2, 24

B

B (begin), subprocedures and, 34
%BIN, 1
bind-by-copy, 57–58, **57**, 60
bind-by-reference in, 58–59, **59**, 60
binding, 56–60
 activation groups and, 66–67
 API interfacing and, 64
 bind-by-copy in, 57–58, **57**, 60
 bind-by-reference in, 58–59, **59**, 60
 binding directory for, 62–63
 in C vs. RPG IV, 26–27
 CEEHDLR API and, 64–65
 CEEHDLU API and, 64
 Create Bound Program (CRTBNDRPG) and, 56, 57, 62
 Create Program (CRTPGM) and, 56, 57, 60
 Create RPG Module (CRTPGMOD) and, 57
 Create Service Program (CRTSRVPGM) and, 58
 prototyping and, 52
 service program signatures and, 60–62
 Update Module (UPDPGM) and, 60
binding directory, QC2LE, 26, 62–63
bit inversion, 90
%bitand, 88–89
%bitnot, 90
%bitor, 91–92
bitwise AND. *See* %bitand
bitwise NOT. *See* %bitnot
bitwise OR. *See* %bitor
bitwise XOR. *See* %bitxor
%bitxor, 92–93
blank lines in code, 78
Bnddir(), 26
built-in functions
 history and development of, 1–2
 uses for, 7–8
bytes used. *See* %size

C

C functions, 19–31
 binding in, RPG IV vs. 26–27
 character strings differences, vs. RPG IV, 26

 data conversion using, 27
 data types and, C vs. RPG IV, 24–25, 25*t*
 exponentiation using, 30–31
 parameter passing to, 25–26
 random numbers from, 22–24
 RPG IV and, how they work, 21–22
 trigonometric, 24
 uses for, 19–21
calculations in subprocedures, 35
CALL, 50
CALLER* activation group, 68
CallP, 21, 34, 50, 54
calls
 to APIs, 63–65
 CALLVL* scoping, 68
 dynamic, 50
CALLVL* scoping, 68
capitalization and case sensitivity, 83
CEEHDLR API, 64–65
CEEHDLU API, 64
CEETREC API, 69
%char, 74, 81, 93–94
 character return data type and, 15*t*
character conversion to graphic. *See* %graphic
character data types, 24
 convert to (%char), 93–94
 converted to unsigned integer (%uns, %unsh), 156
 insert/remove in string (%replace), 134–135
 remove leading/trailing (%trimxx), 154
 translate (%xlate), 158
character return data type, 15, 15*t*
character strings. *See* character data types; strings
character substring, set/retrieve (%subst), 148–149
check field
 left to right (%check), 94–95
 right to left (%checkr), 96
%check, 81, 94–95
 numeric return data type and, 13*t*
%checkr, 81, 96
 numeric return data type and, 13*t*
CL, 1
code changes at positions 1–5, 78

commenting code, 77–78, 83

Const keyword, procedure interface (PI),
prototyping, and, 55

Control Language (CL), 1

control specifications. *See* H control
specification

conversion

- C functions for, 27
- character to double-byte graphic (%graphic),
118
- character/numeric expression to unsigned
integer (%uns, %unsh), 156
- date to date data type (%date), 96–98
- field/expression to packed decimal (%dec,
%dech), 98–99
- negative sign and numeric-to-character
(%editc), 106–107, 107*t*
- numeric to character, using edit codes
(%editc), 104–107, 105*t*, 107*t*
- numeric to character, using edit word
(%editw), 108–110
- numeric to character, using floating-point
notation (%editflt), 107–108
- numeric to floating point (%float), 116
- numeric/character expression to time
(%time), 149–150, 149*t*
- numeric/character expression to timestamp
(%timestamp), 151
- string to UCS-2 varying-length string
(%ucs2), 155
- to character (%char), 93–94
- tokens, Strtok() and, 27, 28–30

Cos/Cosh, 24

Create Bound Program (CRTBNDRPG), 56,
57, 62

- activation groups and, 67, 68

Create Program (CRTPGM), 56, 57, 60

- activation groups and, 67, 68

Create RPG Module (CRTPGMOD), 57

Create Service Program (CRTSRVPGM), 58,
61, 62

- activation groups and, 67, 68

CTDATA**, 79

D

data structures

- elements in, retrieve (%elem), 111
- key (%kds), 121–122
- occurrence number of, set/retrieve (%occur),
128–129

data types

- C vs. RPG IV, 24–25, 25*t*
- character return, 15, 15*t*
- date, time, and timestamp return, 14, 14*t*
- indicator return, 10–12
- numeric return, 12–14
- pointer return, 16, 16*t*
- unsigned integer, 81

DATCV67 module, 52–53

%date, 38, 74, 96–98

- date, time, and timestamp return data types
and, 14, 14*t*

Date-conversion (Date6toLJ) subprocedure for,
37–40

Date6toLJ subprocedure, 37–40

dates

- convert to date data type (%date), 96–98
- date, time, and timestamp return data types
and, 14, 14*t*
- Date-conversion (Date6toLJ) subprocedure
for, 37–40
- day calculation in (%days), 103–104
- difference between (%diff), 101–102
- duration BIFs for, 103–104
- duration limit values, 101*t*
- format conversions and calculations on, 82
- formats for, codes, 97*t*
- month calculation in (%months), 103–104
- prototyping and, conversion using, 52–53
- retrieve component of (%subdt), 146–147
- year calculation in (%years), 103–104

day calculation. *See* %days

%days, 103–104

- no return value in, 17*t*

%dec/%dech, 38, 74, 98–99

- numeric return data type and, 13*t*

decimal position, retrieve. *See* %decpos

`%decpos`, 100
 definition specifications using, 9
 numeric return data type and, 13*t*, 13
definition specifications (D), 79
 `%addr` in, 9
 BIFs in, 8–10
 `%decpos` in, 9
 `%elem` in, 10
 `%len` in, 8–9
 `%paddr` in, 9–10
 `%size` in, 8
 subprocedures and, PI, 34
`%diff`, 101–102
difference between fields. *See* `%diff`
`%div`, 102–103
 numeric return data type and, 13*t*
division. *See* `%div`
duration BIFs, 103–104
duration limit values, 101*t*
dynamic calls, 50
 to APIs, 63–65
dynamic storage allocation. *See* `%realloc`

E

edit codes, 104–107. *See also* `%editc`
edit words, 108–110. *See also* `%editw`
`%editc`, 104–107, 105*t*, 107*t*
 character return data type and, 15*t*
`%editflt`
 character return data type and, 15*t*
`%editw`, 108–110
 character return data type and, 15*t*
`%elem`, 111
 definition specifications using, 10
 numeric return data type and, 13*t*
elements in array/data structure, retrieve. *See*
 `%elem`
embedded SQL, 72
end of file. *See* `%eof`
`%eof`, 4, 112
 filenames on, 81
 indicator data type and, 12*t*
equal status. *See* `%equal`
`%equal`, 113
 indicator data type and, 12*t*

`Setxx` operations and, 82
error condition status. *See* `%error`
`%error`, 114
 indicator data type and, 12*t*
exclusive OR. *See* `%bitxor`
exponentiation using C functions, 30–31
expressions
 extract signed-integer portion from (`%int`,
 `%inth`), 120–121
 set/retrieve length (`%len`), 123
EXTPGM, 52
extract signed-integer portion. *See* `%int`, `%inth`

F

field check
 left to right (`%check`), 94–95
 right to left (`%checkr`), 96
fields
 extract signed-integer portion from (`%int`,
 `%inth`), 120–121
 to update (`%fields`), 115
`%fields`, 115
 no return value in, 17*t*
`%fields` option, 81
file operations
 end of file (`%eof`), 112
 open file check (`%open`), 129
file status. *See* `%status`
filenames on `%eof` and `%found`, 81
`%float`, 116
 numeric return data type and, 13*t*
floating point, 24
 numeric conversion to (`%float`), 116
 numeric to character, floating-point notation
 (`%editflt`), 107–108
format date codes, 97*t*
`%found`, 117
 filenames on, 81
 indicator data type and, 12*t*
free- vs. fixed-format, prototyping and, 50–51
free-format RPG IV, 3, 7–18, 72, 77–84
functions
 defined, 1–2
 vs. op codes, 2–3
 writing your own, 33. *See also* subprocedures

G

global variables, 2
 global visibility concepts,
 subroutines/subprocedures, 33
 graphic character conversion. *See* %graphic
 %graphic, 118

H

H control specification, 26–27, 82
 activation group and, 67, 68
 binding and, example of, 26–27
 half-adjust packed decimal (%dech), 98–99
 %handler, 118–119
 no return value in, 17*t*
 hours calculation. *See* %hours
 %hours, 103–104
 no return value in, 17*t*

I

If, 4
 indicator data type and, 11
 ILE
 binding and, 56–60
 prototyping and, 49–52
*ILE C/C++ Runtime Library Functions
 Reference*, 21
 indicator return data type, 10–12
 indicators, 80
 %int, %inth, 120–121
 numeric return data type and, 13*t*, 13
 integer data types, 24
 Integrated Language Environment. *See* ILE
 inversion, bit. *See* %bitnot

J

JOBLVL* override scoping, 69

K

%kds, 121–122
 no return value in, 17*t*
 key data structure (%kds), 121–122
 keywords and specifications, 82

L

left to right field check (%check), 94–95
 %len, 74, 123
 definition specifications using, 8–9
 numeric return data type and, 13*t*
 length, set/retrieve. *See* %len
 linking, 37
 LOOKUP, 7
 %lookupxx, 4, 5, 124–126
 numeric return data type and, 13*t*
 table type (%tlookupxx), 152–153

M

main procedures and subprocedures
 no shared source members in, 43–47
 NOMAIN and, 44, 58
 shared source members in, 40–43, **41**, **43**
 spec keyword and, 44
 milliseconds calculation. *See* %mseconds
 minutes calculation. *See* %minutes
 %minutes, 103–104
 no return value in, 17*t*
 modular coding, 80, 71–76
 analyzing program needs in, 71–73
 binding and, 56–60
 breaking down programs for, 72
 embedded SQL and, 72
 free-format RPG IV in, 72, 77–84
 gathering code for, 73
 help and collaborative efforts in, 72–73
 nested BIFs and, 74–76
 peer review of code and, 73
 recommended practices in, 82–83
 standards for, 72, 77–84
 structured programming techniques and, 80
 subprocedures in, 72
 Monitor/Endmon, 38
 Monitor/On-error/Endmon, 82–83
 month calculation. *See* %months
 %months, 103–104
 no return value in, 17*t*
 Move/MoveA/MoveL, 7
 %mseconds, 103–104
 no return value in, 17*t*

N

naming conventions, 79–80
activation group, 67
prototyping and, 51–52
special characters used in naming and, 84
negative sign and numeric-to-character
conversion (%editc), 106–107, 107*t*
nested BIFs, 74–76, 82
NEW* activation group, 67, 68
NOMAIN, 44, 58
NoPass* keyword, 55
NOT, bitwise. *See* %bitnot
null indication. *See* %nullind
null-terminated data types, 24
set/retrieve (%str), 142–143
%nullind
indicator data type and, 12*t*
numeric data type
to character conversion, using edit codes
(%editc), 104–107, 105*t*, 107*t*
to character conversion, using edit word
(%editw), 108–110
to character conversion, using floating-point
notation (%editflt), 107–108
to floating point conversion (%float), 116
numeric return data type, 12–14

O

%occur, 128–129
numeric return data type and, 13*t*
occurrence number of data structure,
set/retrieve. *See* %occur
Omit* keyword, 55
open file check. *See* %open
%open, 129
indicator data type and, 12*t*
operation codes/op codes vs. functions, 2–3
operation extender (e), 81, 82–83
Options(*NoPass) keyword, procedure interface
(PI), prototyping, and, 55
Options(*Omit) keyword, procedure interface
(PI), prototyping, and, 55
Options(*RightAdj) keyword, procedure
interface (PI), prototyping, and, 56

Options(*String) keyword, 26
procedure interface (PI), prototyping, and, 56
Options(*Trim) keyword, procedure interface
(PI), prototyping, and, 56
Options(*Varsize) keyword, procedure interface
(PI), prototyping, and, 56
OR, bitwise. *See* %bitor
Original Program Model (OPM) programs and
activation groups, 66
overrides, 68–69

P

P specification, subprocedures and, 34
packed decimal data types, 25
convert field/expression to (%dec, %dech),
98–99
%paddr, 130
definition specifications using, 9–10
pointer return data type and, 16*t*
parameter passing
by reference, 25–26, 51
by value, 26
prototyping and, 51–52
to C functions, 25–26
parameters, 2, 4–6
count of (%parms), 131
number of, limit to, 5–6
prototyping and, 50, 52
subprocedures and, 34
PARM, 50, 51
%parms, 131
numeric return data type and, 13*t*
parse request handler, XML. *See* %handler
passing by reference, 25–26, 51
passing by value, 26
peer review of code, 73
PLIST, 51
pointer return data type, 16, 16*t*
powers. *See* exponentiation
procedure address, retrieve. *See* %paddr
procedure interface (PI)
prototyping and, 51–56
subprocedure definition specs and, 34
procedures, 2
built-in functions in, by return value, 10–16

Program Development Manager (PDM), 56
 program status. *See* %status
 prologues to code, 78
 prototyping, 49–52
 activation groups and, 66–67
 binding and, 52
 CALLP and, 50
 Const keyword in, 55
 date conversion example of, 52–53
 dynamic call and, 50
 dynamic calls to APIs and, 63–65
 free- vs. fixed-format in, 50–51
 keywords for, 54–56
 naming conventions and, 51–52
 Options(*NoPass) keyword in, 55
 Options(*Omit) keyword in, 55
 Options(*RightAdj) keyword in, 56
 Options(*String) keyword in, 56
 Options(*Trim) keyword in, 56
 Options(*Varsize) keyword in, 56
 parameter passing and, 51–52
 parameters and, 50, 52
 passing variables in, 50
 procedure interface (PI) and, 51–56
 string replacement example of, 53–54
 Value keyword in, 54–55

Q

QC2LE binding directory, 26, 62–63
 QCMDEXC API, 63

R

Rand, 20, 22–24
 random number generation, C functions and, 22–24
 %realloc, 131–132
 pointer return data type and, 16*t*
 Reclaim Activation Group (RCLACTGRP), 69
 recommended coding practices, 82–83
 records
 check for existence of character field,
 Typecheck subprocedure for, 35–37
 found status of (%found), 117

%rem, 132–133
 numeric return data type and, 14*t*
 remainders. *See* %rem
 remove leading/trailing characters. *See* %trimxx
 %replace, 74, 75, 81, 134–135
 character return data type and, 15*t*
 Return, 35
 return values, 2, 3–4, 10
 built-in functions and, in procedures, 10–16
 character return data type and, 15, 15*t*
 date, time, and timestamp return data types
 and, 14, 14*t*
 indicator return data type, 10–12
 none, BIFs with, 17, 17*t*
 numeric return data type and, 12–14
 pointer return data type and, 16, 16*t*
 right to left field check. *See* %checkr
 RightAdj* keyword, 56
 RPG IV, 1, 2
 binding and, 26, 56–60
 C language and functions in, 19–21
 data types and, vs. C, 24–25, 25*t*
 exponentiation in, 30–31
 prototyping and, 49–52
 strings in, vs. C, 26

S

scan for characters in string. *See* %scan
 %scan, 136–137
 numeric return data type and, 14*t*
 SCANREPL module, 53–54
 scoping resources with OVRSCOPE, 68–69
 seconds calculation. *See* %seconds
 %seconds, 103–104
 no return value in, 17*t*
 Select/When, indicator data type and, 11
 service programs
 create (CRTSRVPGM), 58, 61, 62
 signature for, 60–62, 60
 update (UPDSRVPGM), 60
 Setxx, %equal and, 82
 %shtdn, 137
 indicator data type and, 12*t*
 shutdown. *See* %shtdn
 signatures, service programs, 60–62

- signed-integers, extract portion from
 - field/expression (%int, %inth), 120–121
- Sin/Sinh, 20, 21, 24
- %size, 138–140, 138*t*
 - definition specifications using, 8
 - numeric return data type and, 14*t*
- source members in subprocedures, 40–43, **41**, **43**
- source members outside of subprocedures, 43–47
- spec keyword, 44
- special characters used in naming, 84
- SQL, 72
- %sqrt, 141
 - numeric return data type and, 14*t*
- square roots, 30. *See also* %sqrt
- Srand, 20, 22–24
- %SST, 1
- standards for programming, 72, 77–84
 - accumulation operators and, 83
 - blank lines in, 78
 - capitalization and case sensitivity in, 83
 - character string manipulation in, 81
 - code changes at positions 1–5, 78
 - comments and, 83
 - comments in, 77–78
 - date and time operations, 82
 - definition specifications (D) in, 79
 - filenames on %eof and %found in, 81
 - H control specifications and, 82
 - indicators in, 80
 - keywords and specifications, 82
 - modular programming techniques and, 80
 - Monitor/On-error/Endmon in, 82–83
 - naming conventions and, 79–80
 - nested BIFs and, 82
 - operation extender (e) and, 81, 82–83
 - prologues to code in, 78
 - recommended practices in, 82–83
 - Setxx operations and, 82
 - special characters used in naming and, 84
 - structured programming techniques and, 80
 - unsigned integer data types and, 81
 - %update operations and fields option, 81
- static-bound modules vs. subprocedures, 34
- %status, 142
- storage
 - activation groups and, 65–70, **65**
 - address in (%addr), 86–87
 - allocation (%alloc), 88
 - dynamic, allocation of (%realloc), 131–132
- storage address. *See* %addr
- %str, 24, 142–143
 - character return data type and, 15*t*
- string token (Strtok), 20
- String* keyword, 56
- strings, 24
 - C vs. RPG IV, 26
 - character substring, set/retrieve (%subst), 148–149
 - convert string to UCS-2 varying-length string (%ucs2), 155
 - data conversion in, using C functions, 27
 - found status of (%found), 117
 - insert/remove characters in (%replace), 134–135
 - manipulation in, 81
 - null-terminated character, set/retrieve (%str), 142–143
 - prototyping and, replacement using, 53–54
 - remove leading/trailing characters (%trimxx), 154
 - scan for characters (%scan), 136–137
 - substrings in array (%subarr), 144–145
 - tokens in, Strtok(), 27, 28–30
- Strtok(), 20, 27, 28–30
- structured programming techniques, 80
- %subarr, 144–145
- %subdt, 146–147
- subprocedures, 2, 33–47
 - activation groups and, 66–67
 - anatomy of, 34–35
 - B (begin) for, 34
 - calculations in, 35
 - code sources for, 73
 - data items in, local nature of, 34–35
 - Date-conversion example (Date6toLJ) of, 37–40
 - global visibility concepts and, 33
 - NOMAIN and, 44, 58
 - P specification for, 34
 - parameters in, 34

PI and definition specifications for, 34
 source members in, 40–43, **41**, **43**
 source members outside of, 43–47
 spec keyword and, 44
 standards for programming, 72
 static-bound modules vs., 34
 subroutines vs., 33–34
 Typecheck example of, 35–37

subroutines
 defined, 2
 global visibility concepts and, 33
 subprocedures vs., 33–34, 33

%subst, 7, 74, 148–149
 character return data type and, 15*t*

substrings in array. *See* %subarr

sum of elements in array. *See* %tfoot

system shutdown. *See* %shdn

T

table lookups. *See* %tfootxx, 152–153

Tan/Tanh, 24

%this, pointer return data type and, 16*t*

time
 convert numeric/character expression to (%time), 149–150, 149*t*
 date, time, and timestamp return data types and, 14, 14*t*
 difference between (%diff), 101–102
 duration BIFs for, 103–104
 duration limit values, 101*t*, 101
 format codes for, 149–150, 149*t*
 format conversions and calculations on, 82
 hours calculation in (%hours), 103–104
 milliseconds calculation in (%mseconds), 103–104
 minutes calculation (%minutes) in, 103–104
 retrieve component of (%subdt), 146–147
 seconds calculation in (%seconds), 103–104

%time, 149–150, 149*t*
 date, time, and timestamp return data types and, 14, 14*t*

%timestamp, 76, 151
 date, time, and timestamp return data types and, 14, 14*t*

timestamps
 convert numeric/character expression to (%timestamp), 151
 date, time, and timestamp return data types and, 14, 14*t*
 difference between (%diff), 101–102
 duration BIFs for, 103–104
 duration limit values, 101*t*

%tfootxx, 152–153
 indicator data type and, 12*t*, 12

tokens, Strtok(), 27, 28–30

totals. *See* %tfoot

translate characters. *See* %xlate

trigonometric C functions, 24

Trim* keyword, 56

%trimxx, 75, 154
 character return data type and, 15*t*

Typecheck subprocedure example, 35–37

U

UCS-2 varying-length string (%ucs2), 155

%uns, %unsh, 156
 numeric return data type and, 14*t*, 14

unsigned integer data types, 81. *See also* %uns, %unsh

update fields, specifying. *See* %fields

Update Module (UPDPGM), 60

%update operations and fields, 81

Update Service Program (UPDSRVPGM), 60

V

validity checking, 5

Value keyword, procedure interface (PI), prototyping, and, 54–55

variables
 bytes used by (%size), 138–140, 138*t*
 global, 2

Varsize* keyword, 56

X

%tfoot, 157
 numeric return data type and, 14*t*

%xlate, 74, 81, 158
 character return data type and, 15*t*
XML, 10
 document identification (%xml), 159
 parse request handler (%handler), 118–119
%xml, 159
 no return value in, 17*t*
XOR, bitwise. *See* %bitxor

Y

year calculation. *See* %years
%years, 103–104
 no return value in, 17*t*

Z

zoned decimal data types, 25